



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Genesys Rules System Deployment Guide

Creating the GRE Application Object in Configuration Manager

12/19/2025

Creating the GRE Application Object in Configuration Manager

Procedure

To create the application object for GRE in Configuration Manager, do the following:

1. Import the GRE application template into Configuration Manager.

Import the GRE Application Template into Configuration Manager

1. In Configuration Manager, navigate to the `Application Templates` folder.
 2. Right-click the `Application Templates` folder, and select `Import Application Template`.
 3. Browse to the templates folder of the installation CD, and select the appropriate template for your version of Management Framework.
 - For Management Framework 8.1.1, select `Genesys_Rules_Engine.apd..`
 - For Management Framework 8.1 and earlier, select `Genesys_Rules_Engine_Generic_Server.apd..`
- Click OK to save the template.

2. Configure the Rules Engine application.

Configure the GRE Application in Configuration Manager

1. Right-click the `Applications` folder and select `New > Application`.
2. Select the template that you imported in the previous procedure.
3. On the `General` tab, enter a name for the application, such as `Rules_Engine`.
4. On the `Tenants` tab, add the Tenants that will be available to the Rules Engine.

5. On the **Server Info** tab, select the Host on which the application will be installed.
6. Add a default listening port.
7. Add an additional port. This port is the connector port on which the Rules Engine Servlet receives requests:
 - The **ID** value is the name of the Rules Engine web application. The default name of this application is `genesys-rules-engine`.
 - The **Listening Port** is the connector port of the Servlet Container. For example, on Tomcat the default listening port is 8080.
 - The **Connection Protocol** must be `http`.
8. On the **Start Info** tab, enter `x` for each field. These fields are not used, but you must enter some text there in order to save the configuration.
9. On the **Options** tab, configure options. Logging options are as follows:

log

Description	Valid values	Default value	Takes effect
all			
Specifies the outputs to which an application sends all log events. The log output types must be separated by a comma when more than one output is configured. For example: <code>all = stdout, logfile</code>	<ul style="list-style-type: none"> • <code>stdout</code>—Log events are sent to the Standard output (<code>stdout</code>). • <code>stderr</code>—Log events are sent to the Standard error output (<code>stderr</code>). • <code>network</code>—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. Setting the <code>all log level</code> option to the network output enables an application to send log events of the Standard, Interaction, and Trace levels to Message Server. Debug-level log 	<code>stdout</code>	After restart

Description	Valid values	Default value	Takes effect
	<p>events are neither sent to Message Server nor stored in the Log Database.</p> <ul style="list-style-type: none"> memory—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. [filename]—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. 		
expire			
Determines how many log files will be kept on disk. If set, expire specifies the maximum number of log files kept on disk.	Any number	(blank)	After restart
segment			
Determines whether a log output written to file is split in multiple segments. If it is, segment	Any number that represents the log size in megabyte	(blank)	After restart

Description	Valid values	Default value	Takes effect
specifies the maximum size of each segment file.			
standard			
<p>Specifies the outputs to which an application sends the log events of the Standard level. The log output types must be separated by a comma when more than one output is configured. For example:</p> <p><code>standard = stderr, network</code></p>	<ul style="list-style-type: none"> • <code>stdout</code>—Log events are sent to the Standard output (<code>stdout</code>). • <code>stderr</code>—Log events are sent to the Standard error output (<code>stderr</code>). • <code>network</code>— Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. • <code>memory</code>—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. • <code>[filename]</code>—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. 	<code>stdout</code>	After restart
trace (not in application template by default)			

Description	Valid values	Default value	Takes effect
Specifies the outputs to which an application sends the log events of the Trace level and higher (that is, log events of the Standard, Interaction, and Trace levels). The log outputs must be separated by a comma when more than one output is configured. For example: <code>trace = stderr, network</code>	<ul style="list-style-type: none"> • <code>stdout</code>—Log events are sent to the Standard output (<code>stdout</code>). • <code>stderr</code>—Log events are sent to the Standard error output (<code>stderr</code>). • <code>network</code>—Log events are sent to Message Server, which can reside anywhere on the network. Message Server stores the log events in the Log Database. • <code>memory</code>—Log events are sent to the memory output on the local disk. This is the safest output in terms of the application performance. • <code>[filename]</code>—Log events are stored in a file with the specified name. If a path is not specified, the file is created in the application's working directory. 	<code>stdout</code>	After restart
verbose			
Determines whether a log output is created. If it is, specifies the minimum level of log events generated. The log events levels, starting with the	<ul style="list-style-type: none"> • <code>all</code>—All log events (that is, log events of the Standard, Trace, Interaction, and Debug levels) are generated. • <code>debug</code>—The same as <code>all</code>. • <code>trace</code>—Log events of the Trace level and higher (that is, log events of the 	<code>standard</code>	After restart

Description	Valid values	Default value	Takes effect
highest priority level, are Standard, Interaction, Trace, and Debug.	<p>Standard, Interaction, and Trace levels) are generated, but log events of the Debug level are not generated.</p> <ul style="list-style-type: none"> interaction—Log events of the Interaction level and higher (that is, log events of the Standard and Interaction levels) are generated, but log events of the Trace and Debug levels are not generated. standard Log events of the Standard level are generated, but log events of the Interaction, Trace, and Debug levels are not generated. none—No output is produced. 		

10. Configure the options on the Settings tab as follows:

Settings in GRE

Description	Valid values	Default value	Takes effect
deployed-rules-directory (added to application template in 8.1.3)			
Specifies the directory in which to keep the working copy of deployed rule packages. When a package is		/GCTI/logs/GRS_Engine (8.1.3 onwards)	After restart

deployed, a copy of the deployed package is placed here. When the rules engine is restarted, all packages defined in this directory are loaded and made available for execution. Specifying a deployed-rules-directory is recommended. If a value is not assigned to the deployed-rules-directory, the rule packages are placed in the WEB-INF\config sub-directory within the genesys-rules-engine web application directory. At this location the deployed rule packages may be deleted when an updated .war file is deployed.

If you choose to change the default value, ensure that the

path exists and that the application server can write to the specified directory.			
max-number-rule-executions			
The maximum number of rules to be executed during a request. This is used to detect unwanted recursion when sequential-mode is false. If this maximum is reached an error is reported. May be set to -1 to denote no maximum.	Any positive integer or -1	10,000	Next rules execution
sequential-mode			
Indicates whether to run the rules engine in sequential mode. In sequential mode, after the initial data set, no more data can be inserted or modified. This allows for the rules engine to operate in a	true/false	false	On rules deployment

simplified way.			
verify-deployer-address			
Indicates whether to verify the TCP address of the application deploying rules to be that of an associated Genesys Rules Authoring Tool.	true/false	true	Immediately
esp-worker-threads (new in 8.1.2)			
Specifies the maximum number of worker threads available when using the ESP interface to execute rules.	Any positive integer	5	Immediately
load-packages-on-start (new in 8.1.4)			
Indicates whether to load deployed rule packages at application start up. If packages are not loaded at startup (value=false), then a package is loaded on its first execution request.	true/false	true	Immediately
json-hierarchical-driver (new in 8.1.4)			

With value true, the <code>JsonHierarchicalStreamDriver</code> class is used to serialize JSON responses. With value false, the <code>JettisonMappedXmlDriver</code> class is used. The Jettison driver is unaware of the original data type and will try to detect numerical values and omit the quotes, whereas the <code>JsonHierarchicalStreamDriver</code> will maintain the data type.	true/false	false	Immediately
--	------------	-------	-------------

11. Save your changes.